1019-1002

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of: Fultheim, Shai	Application No.: 10/828,465
Filed: 04/21/04	Art Unit: 2128
For: Cluster Based Operating System-Agnostic Virtual Computing System	Examiner: David Silver

DECLARATION UNDER 37 CFR 1.132

I, the undersigned, Guy Tel-Zur, hereby declare as follows:

1) I am making this Declaration in support of the patentability of the claims in U.S. Patent Application 10/828,465 (referred to hereinafter as "the Application"). Specifically, this Declaration will set forth my opinion, with supporting facts, that the references that the Examiner has cited against the claims in the Application could not have led a person of ordinary skill in the art to make the claimed invention.

2) I am not an employee of ScaleMP (the assignee of the Application) and have no economic interest in the company. ScaleMP is compensating me for the time I have spent in preparing this Declaration.

3) I have worked in the computing field for the past 14 years, specializing in high-end and high-performance computing. I received my B.Sc. in physics in from Tel Aviv University in 1988,

and my M.Sc. and Ph.D. in physics from the Weizmann Institute of Science (Rehovot, Israel), finishing in 1996. Since that time I have been a researcher and project head in the field of Scientific Computing in the Physics Department of the Nuclear Research Centre Negev (NCRN) in Israel. I am also a visiting scientist in the Electrical and Computer Engineering Department of Ben Gurion University (Beersheba, Israel) and a consultant in high-end computing. My detailed *curriculum vitae* is attached hereto as Exhibit A.

4) I am considered to be an industry authority in the field of high-end computing and have a detailed, first-hand understanding of this field. I am Chairman of the Israeli Association of Grid Technologies (IGT). From 2008-2009, I served as Technology Manager of ASAEL, a consortium of Israeli companies for developing and deploying High-Performance Computing and Cloud Computing technologies. I have also spoken at and organized various conferences on high-end computing technologies, as detailed in my *curriculum vitae*. In the course of my contacts in the field and my work as a project head, I have become familiar with the level of ordinary skill in this area.

5) I have read the Declaration submitted by Dr. Joseph Landman in this Application last year, and I endorse and agree with his conclusions, regarding both the distinction of the invention claimed in the Application over the prior art and the long-felt need in the market for ScaleMP's product, which embodies the invention claimed in the Application. In the present Declaration, I will further demonstrate the distinction of the claimed

invention over the prior art, and particularly over the references cited in the Official Action of December 30, 2009, including Okamoto et al. (U.S. Patent 5,829,041) and the VMware Workstation User's Manual.

6) Computer Virtualization means hiding various kinds of "physical" computing resources from the user by, instead, giving him or her "logical" computing resources. This approach has numerous technological and economical advantages. VMware has long been a market leader in this field, and gives a good summary of the "History of Computer Virtualization" on its Web site (http://www.vmware.com/virtualization/history.html).

7) On this same Web page, VMware lists the advantages users can obtain from VMware products (which I quote directly):

- Run multiple operating systems on a single computer including Windows, Linux and more.
- Let your Mac run Windows creating a virtual PC environment for all your Windows applications.
- Reduce capital costs by increasing energy efficiency and requiring less hardware and increasing your server to admin ratio
- Ensure your enterprise applications perform with the highest availability and performance
- Build up business continuity through improved disaster recovery solutions and deliver high availability throughout the datacenter

- Improve enterprise desktop management & control with faster deployment of desktops and fewer support calls due to application conflicts

Not a single one of these advantages relates to High-Performance Computing, or even parallel computing or cluster computing. The conclusion to be drawn from this lacuna is that even today (not to mention in 2004, when the Application was filed), the very authors of the VMware User Manual cited by the Examiner do not regard their own product as a possible solution for the type of computing configuration described and claimed in the Application, in which the resources of multiple, independent computers are grouped to provide a single, more powerful virtual machine.

8) Virtualization may take place at various different levels. For example Linux offers both user-mode and kernel-mode virtualization, but both of these operating modes are limited to the physical resources of a single "box." On the other hand, the type of virtualization that is described and claimed in the Application takes place at a deeper level, which allows two or more separate computers to be combined under a single virtual machine. The claimed solution aggregates multiple computers into one big computer, as opposed to other virtualization technologies, such as VMware, which partition a single computer into a number of smaller, weaker sub-systems.

9) During the second half of the 1990's, High-Performance Computing systems were developed based on the LOBOS (Lots of Boxes On the Shelf) concept. The idea was to use standard computers and a standard Ethernet interconnect between them, with a separate

operating system (Linux or Unix, for instance) running on each one. In order to perform distributed computing applications, each computer runs a certain share of the application tasks, while accessing data in its own memory space. This is the type of distributed system that Okamoto describes. Such distributed computing techniques enable many processors to cooperate together in order to execute a single job. It is beyond the capabilities of such systems, however, to turn the many computers into a single virtual computer, with a shared memory and a single operating system.

10) ScaleMP has developed a software technology known as the Versatile SMP (vSMP) architecture, which is described and claimed in the Application. This technology is implemented in ScaleMP's vSMP Foundation software products. It provides a shared virtual machine across multiple independent physical machines. vSMP software aggregates resources of the underlying physical hardware, providing a single process space, memory address space, and I/O space for an operating system. As a result, multiple physical machines operate as a single virtual machine with the aggregated number of CPUs, memory size, and I/O space. This is a different, novel, and unique form of virtualization of resources, relative to methods that were previously known for the X86 architecture. VMware takes the opposite approach, subdividing a single physical machine into many smaller "virtual machines."

11) Turning now to the specifics of the cited references, Okamoto makes very clear, even in his Abstract, that each computer

in his distributed system retains its own independent operating system, memory space, and program flow:

> "...The single virtual space for arranging programs and/or data among a plurality of computers forming the distributed system are divided into a plurality of regions called memory chapters, and a part of the single virtual space to be managed independently by each computer is requested from each computer in units of these memory chapters. Then, a server allocates one of the memory chapters to each computer in response to each request from each computer, while managing allocations of the memory chapters to the plurality of computers so as not to allocate each one of the memory chapters to more than one computers." (emphasis added)

In other words, although available physical memory space may be distributed among multiple hosts on the network, Okamoto's server ensures that each host gets its own distinct memory allocation. Each host maintains its own memory image, and each "chapter" is protected in such a way that only the allowed host can write or overwrite its content. This may be a good idea for sandboxing and security, but it is diametrically opposed to the shared memory structure that is needed to support a shared virtual machine.

12) According to the VMware manual (page 13), a virtual machine "is equivalent to a PC with a unique network address and a full complement of hardware devices." In other words, a virtual machine, as the term is understood in the computing field, appears to its guest operating system and to applications as a single computer. In a single computer, the CPU (or CPUs, in the case of

a multiprocessor) is able to access all resources of the computer, including the entire available memory space.

13) The Examiner suggests in the Official Action of December 30 that Okamoto's operating systems, running on the host computers on his network, function as virtual machine implementers. This statement in itself is questionable, since Okamoto has nothing like a virtual machine in the sense in which this term is normally used, which I have explained above. If there is no virtual machine (or even a suggestion of a virtual machine) in Okamoto, what use would Okamoto have for a virtual machine implementer, and what could have led a person of ordinary skill in the art to ccnsider Okamoto's operating system to be anything other than an operating system? In fact, Okamoto's operating systems are just that - host operating systems, with added capabilities for using distributed memory. Perhaps the Examiner was confused by Okamoto's use of the term "virtual space" in reference to memory mapping. A person of ordinary skill in the art would have understood that there is no relation between "virtual memory" and a "virtual machine."

14) A person of ordinary skill in the art, reading the VMware manual, would see (pages 14-15) that in order to run VMware, he must first have a computer, and the computer must run a host operating system, like Okamoto's operating system. Only when these elements are in place, can the virtual machine implementer (the VMware Workstation described in the manual, commonly referred to as a hypervisor) be installed over the host operating system. Therefore, this person of ordinary skill might have considered

installing the VMware Workstation over the Okamoto operating system, but he would never have considered Okamoto's operating system to be a VMware Workstation, as the Examiner appears to believe. This person might install multiple VMware Workstations on multiple computers in Okamoto's network, but the resulting virtual machines would simply remain multiple, separate virtual machines, each running on its own host.

15) The VMware Workstation is not just an application program. An application program does not have "a unique network address and a full complement of hardware devices," which are characteristics of the virtual machine described in the VMware manual. In order to attain these characteristics, it was necessary for VMware (according to their Web page that I cited above) to develop "an adaptive virtualization technique that 'traps' [host hardware] instructions as they are generated and converts them into safe instructions that can be virtualized." As I have explained above, VMware exercises this capability only on a single host machine.

16) In other words, the VMware Workstation itself is a single-computer program, and VMware makes clear that getting the program to run on a single computer was no easy accomplishment. Against this background, the Examiner makes the contradictory assertion that VMware could run over Okamoto's distributed computer system like any other application. This assertion contradicts what VMware itself states. Based on the state of the art in the field of virtual computers and multiprocessor systems, a person of ordinary skill would have reached the opposite

conclusion: that adapting the VMware Workstation to run over multiple separate computers would be a difficult, if not impossible, task. Neither VMware nor Okamoto (nor any other reference of which I am aware prior to the Application) gives any suggestion of a solution to the problems of virtualizing instructions and presenting a single, shared view of computer resources over a multi-computer system. Neither Okamoto nor VMware provides the tools necessary for this sort of implementation.

17) For the sake of completeness, I studied the other references that were cited by the Examiner in the December 30 Official Action. Specifically, Altman et al. (US 2004/0054517) deals with porting a parallel program from one multi-processor computer to another multi-processor computer (a host) in-order to execute the parallel program on that computer. Although Altman mentions virtualization, he says nothing about sharing any sort of virtual machine between the computers, and therefore does not provide any of the missing knowledge needed to arrive at the invention that is claimed in the Application. Dr. Landman has already explained why Nickel et al. (US 2003/0005068) would not have been useful in this regard, and the other references are no more relevant.

18) As an expert in the field of High-Performance Computing, I can state unequivocally that solving the technical problems of running a shared virtual machine over multiple separate computers is a challenging task, which persons of ordinary skill in the art would have difficulty in completing today, let alone in 2004. The

methods, software and systems described in the Application represented a major breakthrough over the state of the art at the time. I endorse Dr. Landman's finding that the invention by ScaleMP of this sort of shared virtual machine is a surprising result, which answers a long-felt need in my field.

18) Thus, to conclude, a person having ordinary skill in the art would not have been motivated or able to create the invention that is described and claimed in the Application from the cited references, and specifically not from Okamoto and VMware.

19) I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and conjecture are thought to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application of any patent issued thereon.

Guy Tel-Zur, Ph.D.

Dr. Guy Tel-Zur, Citizen of Israel
8 Sigalon Street, Omer, Israel

Date: 4/22/2010

Exhibit A: Curriculum Vitae of Guy Tel-Zur

Specialties:

Parallel Processing, Grid Computing, Cloud Computing, High-Performance Computing, High-Throughput Computing, Linux Clusters, Scientific Computing, Computational Physics, Consulting, Technology Forecasting, Experimental High-Energy Physics, Purchasing, Scientific Writing

Education:

1991 - 1996 Ph.D. - Physics Dept., The Weizmann Institute of Science, Israel.

Thesis title: *Electron Pair Production in p-Be and p-Au collisions at 450GeV/c.*

1988 - 1990 MSc. - Physics Dept., The Weizmann Institute of Science, Israel.
1985 - 1988 BSc. - Physics Dept., Tel Aviv University, Israel.

Professional Experience:

1996-Present
Researcher and project head in the field of Scientific Computing, Physics Dept., NCRN, Israel.

2000-Present
External Lecturer and Researcher, Parallel and Distributed Computing, Ben-Gurion University of the Negev (BGU), Israel.

2004-Present
Consultant in High-End Computing.

2008-Present
Chairman of the Israeli Association of Grid Technologies (IGT) (www.Grid.org.il).

2008-2009
Technology Manager at ASAEL – A consortium, in the frame work of MAGNET, of several leading Israeli firms for developing and deploying High-Performance Computing and Cloud Computing technologies in industrial process control.

Teaching Experience:

2002-Present

Teaching "*An Introduction to Parallel Processing*" course at the Electrical & Computer Engineering Dept., Ben-Gurion University (BGU).

2010- Developing two new courses, which were approved for the next academic year, in Advanced Parallel Processing and in Computational Physics to be taught at the BGU.

Sabbatical:

2003 - 2004 A Visitor Scientist at the Condor team, Computer Science Dept., The University of Wisconsin, Madison, USA.

2009 – 2010 A Visitor Scientist at the Electrical and Computer Engineering Dept. BGU.

Research Fellowship:

1996 - 2002 "A. Katzir" Fellowships from the Israel Ministry of Defense.

Publications:

24 publications (the full list is available at: http://tel-zur.com) and several internal NRCN research reports.

Membership and Affiliations:

- IEEE Computer Society, ACM, SIAM.
- Member of the board of editors of "PhysicaPlus" magazine (http://physicaplus.org.il)
- Member of the board of editors of the Israeli edition of Scientific American (http://sciam.co.il)

Miscellaneous:

- Supervisor to 4th year & MSc. engineering and CS students, BGU
- Reviewer, IEEE Transactions on Computing.
- Conference Co-organizer of the 1st IGT annual event. December 6th 2005.
- Creator of the "Cloud Computing" group in Facebook with more than 3000 members
- Program Committee member at the "Cloud-based Services and Applications" workshop, IEEE eScience2009 Oxford, UK (http://www.oerc.ox.ac.uk/ieee/workshops/cloud)

- Program Committee member at the "CBSA: Cloud-based Services and Applications, IEEE Services2010, Miami Florida, USA. Date: July 5 - 10, 2010